ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ
INFORMATION TECHNOLOGY, COMPUTER SCIENCE, AND MANAGEMENT







УДК 004.01
https://doi.org/10.23947/2687-1653-2022-22-4-384-390

Научная статья



Разработка архитектуры по подключению системного модуля для людей с ограниченными возможностями

А. А. Баскаков✉, **А. Г. Тарасов**
Московский государственный технологический университет «СТАНКИН», Российская Федерация, г. Москва, Вадковский пер., 1
✉ aleks.baskakov@mail.ru

Аннотация
Введение. Для разработки новых системных модулей программного обеспечения помощи сотрудникам с ограниченными возможностями требуется проработать архитектурное решение для взаимодействия всех частей системы. В результате анализа и проектирования необходимо получить программную архитектуру, которая должна выполнять ряд стандартных требований. В первую очередь она должна быть безопасной. Для этого следует учитывать систему логирования ошибок, аудирование событий, возможность отключения функционала непосредственно после вывода в промышленную эксплуатацию, внутренние механизмы валидации входных запросов клиента и ответов сервера. Данная работа посвящена выработке основных вариантов обслуживания системы, анализу исключительных ситуаций при взаимодействии с пользователем для дальнейшей оценки эффективности архитектуры и непосредственной разработки проекта.
Материалы и методы. Архитектурное решение проводилось при помощи языка графического описания (Unified Modeling Language, UML), который помогает строить визуальные изображения жизненного цикла и взаимодействия всех компонентов системы. Для проработки взаимодействия основных модулей будущей системы использовался синтаксис диаграммы развертывания (deployment diagram UML). Для обработки жизненного цикла — синтаксис диаграммы последовательности действий (sequence diagram UML). Помимо этого использовалась диаграмма прецедентов для описания основных сценариев использования.
Результаты исследования. Разработана архитектура, имеющая схему взаимодействия отдельных модулей и систем, а также варианты использования программного комплекса для будущей реализации программного продукта. Предложенная архитектура системы соответствует требованиям безопасности, надёжности (отказоустойчивости) и производительности. Авторами зафиксированы функциональные требования системы помощи сотрудникам предприятий с проблемами слуха для возможности их трудоустройства и работы по телекоммуникационной сети интернет. Выработаны основные вариации обслуживания системы.
Обсуждение и заключения. Построение грамотной архитектуры позволяет учесть ситуации, выходящие за рамки нормального использования системы, а также использовать нечеткую модель для определения эффективности системы. Дальнейшее углубленное описание вариантов развёртывания и эксплуатации позволит реализовать эффективную и производительную систему.

Ключевые слова: диаграмма развертывания UML, диаграмма последовательности UML, архитектура программного обеспечения, коммерческое программное обеспечение, диаграмма прецедентов UML.

Благодарности. Авторы выражают благодарность Когану Юрию Григорьевичу, кандидату технических наук, доценту кафедры Информационных технологий и вычислительных систем МГТУ «СТАНКИН», участвующему в разработке вариантов использования системы в роли эксперта данной области.

Для цитирования. Баскаков, А. А. Разработка архитектуры по подключению системного модуля для людей с ограниченными возможностями / А. А. Баскаков, А. Г. Тарасов // Advanced Engineering Research. — 2022. — Т. 22, № 4. — С. 384–390. https://doi.org/10.23947/2687-1653-2022-22-4-384-390

Original article

Development of Architecture for Connecting a System Module for People with Disabilities

Alexey A. Baskakov ✉, **Alexey G. Tarasov**
Moscow State University of Technology (MSUT "STANKIN"), 1, Vadkovsky Lane, Moscow, Russian Federation
✉ aleks.baskakov@mail.ru

Abstract

Introduction. To develop new system modules of software to help employees with disabilities, it is required to work out an architectural solution for the interaction of all parts of the system. As a result of the analysis and design, it is necessary to obtain a software architecture that must meet a number of standard requirements. First of all, it should be safe. To do this, you should take into account the error logging system, event auditing, the possibility of disabling the functionality immediately after putting it into commercial operation, internal mechanisms for validating client input requests and server responses. This study is aimed at the development of basic system maintenance options, the analysis of exception cases under interacting with the user for further evaluation of the architecture efficiency, and the direct project development.

Materials and Methods. The architectural decision was carried out using the Unified Modeling Language (UML), which helps to build visual images of the life cycle and interaction of all components of the system. The syntax of the UML deployment diagram was used to study the interaction of the main modules of the future system, and the syntax of the UML sequence diagram was used to process the lifecycle. A use case diagram was also applied to describe the main use cases. To study the interaction of the main modules of the future system, the UML deployment diagram syntax was used. For life cycle processing, the UML sequence diagram syntax was applied. In addition, a use case diagram was applied to describe the base use cases.

Results. An architecture that has a scheme for the interaction of individual modules and systems, as well as options for using the software package for the future implementation of the software product, has been developed. The proposed system architecture meets the requirements of security, reliability (fault tolerance), and performance. The authors have fixed the functional requirements of the system of assistance to employees of enterprises with hearing problems for the possibility of their employment and work on the telecommunication Internet. Basic variations of system maintenance have been developed.

Discussion and Conclusions. Building a competent architecture provides taking into account cases that go beyond the normal use of the system, and applying a fuzzy model to determine the system efficiency. Further in-depth description of deployment and operation options will enable to implement an efficient and productive system.

Keywords: UML deployment diagram, UML sequence diagram, software architecture, commercial software, UML use case diagram.

Acknowledgements. The authors would like to thank Yuri G. Kogan, Cand.Sci. (Engineering), associate professor of the Department of Information Technologies and Computing Systems, STANKIN Moscow State Technical University, who participated in the development of the system use cases as a subject matter expert.

For citation. A. A. Baskakov, A. G. Tarasov. Development of Architecture for Connecting a System Module for People with Disabilities. Advanced Engineering Research, 2022, vol. 22, no. 4, pp. 384–390. https://doi.org/10.23947/2687-1653-2022-22-4-384-390

Введение. Исследование причин нетрудоспособности среди населения с инвалидностью ранее проводили путем использования экспертного оценивания методом попарных сравнений Томаса Л. Саати. В результате было выявлено, что для улучшения трудоспособности требуется разработать комплексную систему для решения проблемы потери слуха и возможности работать в центрах удаленной поддержки, где понимание собеседника является обязательным [1]. Перед оценкой эффективности требуется выработать все возможные варианты обслуживания системы. Это позволит определить эффективность сервисной архитектуры, связанной с выполнением каждым модулем нужного уровня обслуживания. Для уменьшения трудозатрат на этапе разработки необходимо предварительное определение исключительных ситуаций при работе модуля с транскрибированием.

Научная новизна данного исследования состоит в эффективности архитектуры относительно решаемых задач (вариантов обслуживания).

Оценка производилась при помощи нечеткой модели мнений эксперта в данной области и выходит за рамки обсуждения данной статьи.

Для разработки архитектуры нужно учитывать следующие параметры:

– кроссплатформенность — программный продукт должен охватывать основные операционные системы;

– отказоустойчивость — система должна стабильно работать в случае перебоев в работе;

– безопасность — система должна поддерживать логирование, мониторинг и аудирование событий пользователя;

– горизонтальную масштабируемость — при увеличении количества клиентов программный продукт должен работать с той же скоростью за счет увеличения количества серверов;

– производительность — система должна работать без задержек в режиме реального времени.

Планирование и описание программной архитектуры — важный и необходимый шаг перед непосредственной разработкой клиент-серверного приложения. Система должна иметь модуль для распознавания речи и переводить ее в текст, тем самым позволяя людям с проблемами слуха работать на удаленной поддержке.

Материалы и методы. UML — это специальный язык моделирования, который применяется при разработке архитектуры вычислительных систем, программного обеспечения, сетевой архитектуры и при построении бизнес-процессов[1] [2, 3].

Для описания архитектуры проекта использовались следующие схемы:

1. Диаграмма последовательности (т. н. sequence diagram) — описывает процесс приема звонков и их обработки с точки зрения сотрудника организации[2].

2. Диаграмма прецедентов (т. н. use case diagram) — описывает варианты использования системы.

3. Диаграмма развертывания (т. н. deployment diagram) — описывает архитектуру системы.

Результаты исследования. Построение UML диаграмм позволило определить критерии качества обслуживания системы, учесть исключительные ситуации и способы их обработки, а построение диаграммы развертывания — определить эффективность системы, используя нечеткую модель.

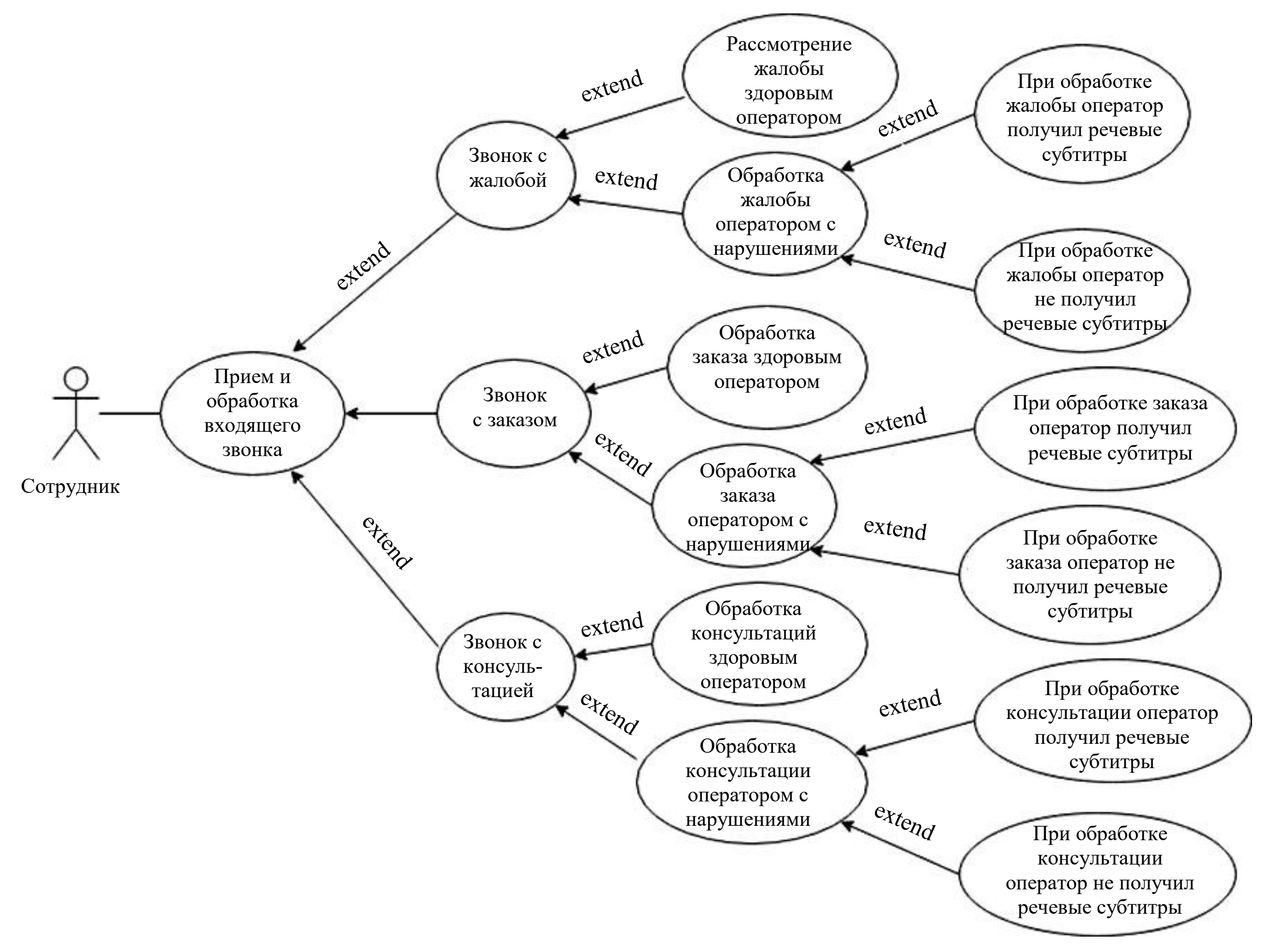


Рис. 1. Диаграмма вариантов использования системы (рисунок авторов)

[1] Хамматова Л. А. Универсальный язык моделирования UML, основные диаграммы и проблемы использования // Прорывные научные исследования: проблемы, закономерности, перспективы : сб. ст. XIII междунар. науч.-практ. конф. Пенза, 2019. С. 88–90.

[2] Тесленко И. Б., Царев А. О. Особенности информационного проектирования с использованием языка UML // Инновационное развитие социально-экономических систем: условия, результаты и возможности : ма-лы V междунар. науч.-практ. конф. Орехово-Зуево, 2017. С. 174–177.

На рис. 1 показана диаграмма прецедентов, разработанная авторами данной статьи, которая показывает, что основные запросы к удаленной поддержке поступают по следующим причинам: жалобы на оказание услуг предприятием, консультация и заказ услуг. При этом стоит учитывать, что система автоматической генерации субтитров может отрабатывать некорректно по некоторым причинам — не смогла распознать речь, не пройдена фильтрация запрещенных слов и т.д. В таком случае следует переключать клиента на роботизированную систему или другого оператора.

Роботизированная система представляет собой модуль автоматической генерации голоса с собственным жизненным циклом, позволяющим проинформировать клиента о технических проблемах во время звонка. Модуль также позволяет по воле клиента переключиться на другого свободного оператора или занять очередь в случае большой нагрузки в данный момент времени.

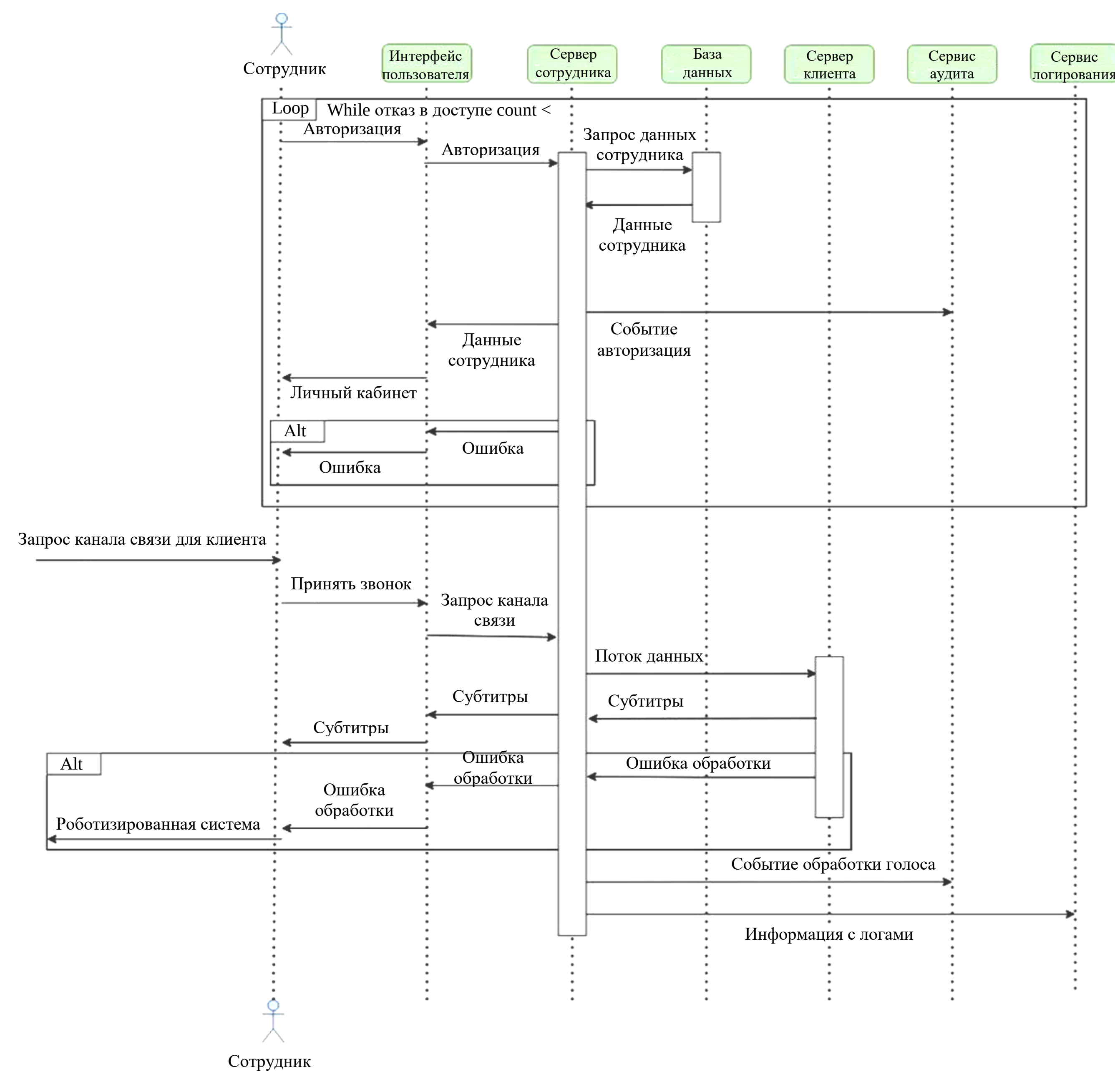


Рис. 2. Диаграмма последовательности действий сотрудника (рисунок авторов)

На рис. 2 показана диаграмма последовательности действий с точки зрения сотрудника организации. После стандартной процедуры авторизации через широковещательный канал в личный кабинет сотрудника может поступить звонок от клиента. В случае принятия звонка, между клиентом и сотрудником будет установлен непрерывный канал передачи информации. Полученные данные в режиме реального времени передаются в модуль обработки, где при помощи набора преобразований и методов генерируются текстовые субтитры.

В случае ошибки преобразования, клиент переключается на роботизированную систему, а сотрудник уведомляется о невозможности обработки голоса. Все действия (события) клиента и сотрудника отмечаются в модуле аудирования и логирования.

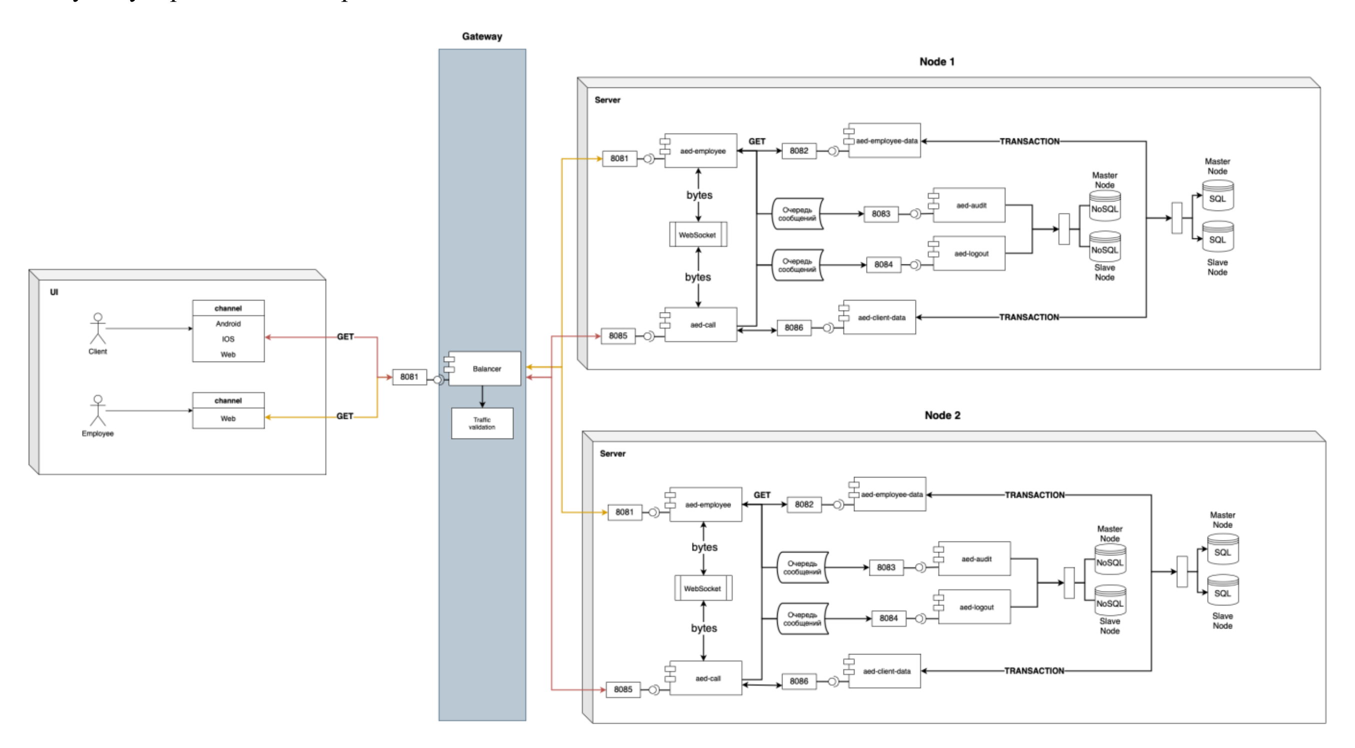


Рис. 3. Диаграмма развертывания компонентов системы (рисунок авторов)

На рис. 3 представлена диаграмма развертывания проекта микросервисной архитектуры [4, 5], которая состоит из трех блоков.

Internet. В сети интернет расположены визуальные модули для взаимодействия пользователя и клиента (User Interface, UI). Стоит заметить, что каналы для реализации у клиента представлены тремя платформами: IOS, Android, Web, тогда как у сотрудника — только Web-каналом. В первую очередь, это связано с тем, что сотруднику обычно не требуется подобные приложения под мобильные операционные системы из-за характера работы и безопасности.

Шлюз передачи данных. Основная задача шлюза — обеспечить передачу информации из сети интернет в локальную сеть, где развернуты основные сервера, и обратно. Это дает ряд преимуществ. Во-первых, появляется общая точка для передачи всех данных. Невозможно взаимодействовать с сервером в обход шлюза безопасности. Во-вторых, все запросы можно проверять на соответствие изначально заложенному формату. В-третьих, шлюз позволят балансировать нагрузку между узлами, тем самым достигается отказоустойчивость. Если один узел перестанет функционировать, все запросы будут автоматически распределены между другими узлами. В-четвертых, систему можно легко масштабировать. Если количество запросов со временем вырастет, достаточно добавить новый узел и производительность останется на прежнем уровне.

Узлы. Это комплексная система, состоящая из сервисов, взаимодействующих между собой. Основные из них — это сервис для получения запросов от клиента и сотрудника. Их задача получить запрос, обработать его и настроить взаимодействие между собой при помощи специальных компонентов — сокетов, которые позволяют обмениваться данными в режиме реального времени. Эти компоненты получают данные о клиенте и сотруднике из специальных сервисов, взаимодействующих с реляционной базой данных при помощи языка SQL [6]. Для отказоустойчивости и скорости база данных также имеет несколько узлов, но не менее двух. Все действия пользователя и клиента передаются в сервисы аудирования и логирования при помощи очередей сообщений. Это дает некоторые преимущества — асинхронность передачи данных и сохранность всей информации в случае отказа сервисов. При этом хранение такой информации нецелесообразно в реляционной базе в связи с однотипностью и простотой структуры. По этой причине желательно использовать документно-ориентированную СУДБ NoSQL [7, 8][3]. Стоит заметить, что независимые базы данных в каждом из узлов

[3] Попов В. Б., Гавриков И. В. Технологии «NOSQL» в алгоритмах анализа больших данных и искусственном интеллекте // Проблемы информационной безопасности : сб. тр. V всерос. с междунар. участием науч.-практ. конф. Симферополь, Гурзуф, 2019. С. 158–160.

обеспечивают высокую производительность, а консистентность и актуальность данных реализуется через механизм, репликации которого будут подробно описаны в следующих исследованиях. Механизм «Master-Slave» в реляционных и нереляционных базах данных необходим для возможного вертикального масштабирования и минимизации затрат ресурсов в случае увеличения нагрузки на отдельный узел.

Все запросы из сети Internet в локальную сеть происходят по протоколу HTTP следующими методами:

– GET — в случае получения информации от любого из модулей;

– POST — в случае создания записей в любом из модулей;

– PUT — в случае изменения в любом из модулей.

Запросы к сохраняемой информации происходят внутри транзакций для обеспечения целостности данных [9–11].

Обсуждение и заключения. Разработка архитектуры, используя язык графического описания, позволяет наглядно описывать функциональные требования к системе, оценивать её эффективность и уменьшать количество ошибок.

Диаграмма вариантов использования позволила, в случае возникновения исключительных ситуаций при транскрибировании голоса, выявить необходимость разработки роботизированной системы, а также определить множество вариантов обслуживания для нечеткой модели. Диаграмма развёртывания помогла определить эффективность выполнения функций каждым внутренним сервисом, а диаграмма последовательности действий — определить жизненный цикл продукта.

Список источников

1. Баскаков, А. А. К проблеме использования автоматизированного рабочего места людьми с ограниченными возможностями / А. А. Баскаков, А. Г. Тарасов // Advanced Engineering Research. — 2021. — Т. 21, № 3. — С. 290–296. https://doi.org/10.23947/2687-1653-2021-21-3-290-296

2. From UML State Machine Diagram into FPGA Implementation / G. Bazydto, M. Adamski, M. Węgrzyn, A. Rosado Munoz // IFAC Proceedings Volumes. — 2013. — Vol. 46. — P. 298–303. https://doi.org/10.3182/20130925-3-CZ-3023.00061

3. Shailaja Uke. UML Based Modeling for Data Aggregation in Secured Wireless Sensor Network / Shailaja Uke, Ravindra Thool // Procedia Computer Science. — 2016. — Vol. 78. — P. 706–713. https://doi.org/10.1016/j.procs.2016.02.120

4. Saulo S.de Toledo. Identifying Architectural Technical Debt, Principal, and Interest in Microservices: A Multiple-Case Study / Saulo S. de Toledo, Antonio Martini, Dag I. K. Sjøberg // Journal of Systems and Software. — 2021. — Vol. 177. — Art. 110968. https://doi.org/10.1016/j.jss.2021.110968

5. Nuno Mateus-Coelho. Security in Microservices Architectures / Nuno Mateus-Coelho, Manuela Cruz-Cunha, Luis Gonzaga Ferreira // Procedia Computer Science. — 2021. — Vol. 181. — P. 1225–1236. https://doi.org/10.1016/j.procs.2021.01.320

6. Григорьев, Ю. А. Оценка времени выполнения сложного SQL-запроса в СУБД MS SQL SERVER 2000 / Ю. А. Григорьев, В. Г. Матюхин // Информатика и системы управления. — 2004. — С. 3–13.

7. Давыдов, Д. А. Тенденции развития NOSQL-СУБД / Д. А. Давыдов, П. С. Манылов // Научно-технический вестник Поволжья. — 2013. — № 3. — С. 131–135.

8. Шарипова, Н. Н. Об использовании NOSQL-хранилищ данных / Н. Н. Шарипова // Wschodnioeuropejskie czasopismo naukowe. — 2016. — Т. 9, № 3. — С. 73–76.

9. Chodak, G. HTTP-Level E-Commerce Data Based on Server Access Logs for an Online Store / G. Chodak, G. Suchacka, Y. Chawla // Computer Networks. — 2020. — Vol. 183. — Art. 107589. https://doi.org/10.1016/j.comnet.2020.107589

10. Okumura, N. Formal Analysis of RFC 8120 Authentication Protocol for HTTP under Different Assumptions / N.Okumura, K. Ogata, Y. Shinoda // Journal of Information Security and Applications. — 2020. — Vol. 53. — Art. 102529. https://doi.org/10.1016/j.jisa.2020.102529

11. Mattson, R. L. R. HTTP-MPLEX: An Enhanced Hypertext Transfer Protocol and Its Performance Evaluation / Robert L. R. Mattson, Somnath Ghosh // Journal of Network and Computer Applications. —2009. — Vol. 32. — P. 925–939. https://doi.org/10.1016/j.jnca.2008.10.001

Поступила в редакцию 10.09.2022.
Поступила после рецензирования 24.10.2022.
Принята к публикации 24.10.2022.

Об авторах:

Баскаков Алексей Андреевич, аспирант кафедры «Информационные технологии и вычислительные системы» Московского технологического университета «МГТУ СТАНКИН» (127055, РФ, г. Москва, Вадковский пер., 1), ORCID, aleks.baskakov@mail.ru

Тарасов Алексей Геннадиевич, доцент кафедры «Информационные технологии и вычислительные системы» Московского технологического университета «МГТУ СТАНКИН» (127055, РФ, г. Москва, Вадковский пер., 1), кандидат технический наук, ORCID, tarasov.ag@mail.ru

Заявленный вклад соавторов:

А. А. Баскаков — проведение расчетов, подготовка текста, поиск научной литературы, формирование выводов. А. Г. Тарасов — научное руководство, формирование цели и задач, корректировки текста.

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.

Все авторы прочитали и одобрили окончательный вариант рукописи.